AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions and listings of claims in the application:

1. (Currently amended) A method in a data processing system containing source code with a subprogram having at least one of an integer non-scalar parameter and a logical non-scalar parameter, the method comprising ~~the steps of~~:

creating an interface file for the subprogram in the source code;

storing in the interface file a definition of the subprogram;

adding to the interface file a directionality of at least one of the integer parameter and the logical parameter based on comments in the source code;

adding to the interface file a parameter size along each dimension of at least one of the integer parameter and the logical parameter

~~a comment for at least one of the integer and logical parameters, the comment indicating the parameter passing at least one of semantics and extent of the dimension along each of the dimensions of a non-scalar parameter~~; and

reading the interface file to generate a stub routine that converts at least one of the integer and logical parameters from 32-bit to 64-bit and that invokes the subprogram by specifying the converted parameters.

2. (Original) The method of claim 1, wherein the source code is 32-bit code and wherein the method further includes the step of:

invoking the 64-bit code from 32-bit code.

3. (Previously presented) A method in a data processing system, comprising the steps of:

receiving 32-bit source code; and

automatically generating a 32-bit to 64-bit conversion stub that is used by the 32-bit source code to invoke 64-bit code.

4. (Original) The method of claim 3, wherein the 32-bit source code has a subprogram with an integer or logical parameter and wherein the automatically generating step further includes the steps of:

creating an interface for the subprogram;

inserting a code-generator statement into the interface describing a characteristic of the parameter; and

using the interface to create a stub for use as a 32-bit to 64-bit converter.

5. (Original) A data processing system, comprising:

a storage device, comprising:

source code with a subprogram having at least one of an integer and logical parameter;

an interface generator that reads the subprogram and that generates an interface file with indications of characteristics of the parameter; and

a stub generator that reads the interface file and that generates a stub for the subprogram by using the characteristics, wherein each of the stubs receives a set of parameter values, generates the values for the required parameters from the received set of parameter values, and invokes the subprogram with the values for the parameters; and

a processor for running the interface generator and the stub generator.

6. (Original) The data processing system of claim 5, wherein the source code contains comments indicating the characteristics of the parameter.

7. (Original) The data processing system of claim 6, wherein the characteristics include an indication of a conditional value for at least one of the required parameters.

8. (Original) The data processing system of claim 6, wherein the characteristics include an indication of whether at least one of the required parameters is used to contain a return value.

9. (Original) The data processing system of claim 6, wherein the characteristics include a directionality of at least one of the required parameters.

10. (Original) The data processing system of claim 6, wherein the characteristics include an indication of whether at least one of the required parameters returns a multidimensional variable.

11. (Original) The data processing system of claim 6, wherein the characteristics include an indication of whether a size of at least one of the required parameters is based on another one of the required parameters.

12. (Original) The data processing system of claim 6, wherein the characteristics include an indication of whether at least one of the required parameters is a work space parameter.

13. (Previously presented) A computer-readable medium containing instructions for controlling a data processing system to perform a method comprising the steps of:

receiving 32-bit source code; and

automatically generating a 32-bit interface to 64-bit source code.

14. (Previously presented) The computer-readable medium of claim 13, wherein the 32-bit source code has a subprogram with a parameter and wherein the automatically generating step further includes the steps of:

creating an interface for the subprogram;

inserting a code-generator statement into the interface describing a characteristic of the parameter; and

using the interface to create a stub for use as the interface to the 64-bit code.

15. (Original) A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having source code with a subprogram having a parameter, the method comprising the steps of:

reading the source code; and

generating a stub routine that invokes the subprogram and that facilitates use of at least one of a converted integer and logical parameter.

16. (Previously presented) A data processing system comprising:

means for receiving 32-bit source code; and

means for automatically generating a 32-bit to 64-bit conversion stub that is used by the 32-bit source code to invoke 64-bit code.